



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/655,161	09/04/2003	Man-Ho Lawrence Lee	200209146-1	6803

22879 7590 05/16/2007
HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER

WU, JIANYE

ART UNIT	PAPER NUMBER
----------	--------------

2609

MAIL DATE	DELIVERY MODE
-----------	---------------

05/16/2007

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

<p align="center">Office Action Summary</p>	Application No. 10/655,161	Applicant(s) LEE, MAN-HO LAWRENCE	
	Examiner Jianye Wu	Art Unit 2609	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☐ Responsive to communication(s) filed on ____.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-24 is/are pending in the application.
- 4a) Of the above claim(s) ____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) ____ is/are allowed.
- 6) ☒ Claim(s) 1-24 is/are rejected.
- 7) ☐ Claim(s) ____ is/are objected to.
- 8) ☐ Claim(s) ____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 11/6/07 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. ____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. ____. |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date ____. | 6) <input type="checkbox"/> Other: ____. |

DETAILED ACTION

1. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. **claim 21** is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

For **claim 21**, the claimed software is non-statutory subject matter since it is not a process, machine, manufacture nor composition of matter; nor it is recorded on some computer-readable medium, see MPEP 2106(IV)(B)(1).

Claim 21 lacks the proper preamble language for statutory computer program product. See MPEP 2100 for guidance on computer related inventions.

The examiner suggests a preamble as follows:

"A computer readable medium encoded with computer executable instructions to perform a method, the method comprising:"

Correction is required.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claim 1-2, 5-10, 12, 17-19, and 20-24 are rejected under 35 U.S.C. 102(b) as being anticipated by Frye et al, IETF RFC 2576 "Coexistence between Version 1,

Version 2, and Version 3 of the Internet-standard Network Management Framework";
March 2000, herein after being referenced as RFC 2576.

For **Claim 1**, RFC 2576 discloses a method for minimizing compatibility issues among interacting components of different dialect versions, comprising:

defining a plurality of type-version identifiers (e.g., messageProcessingModel and securityModel first 2 bullets, Page 27) each indicating a corresponding type and version from among a plurality of types and versions of requests;

installing at least one handler (Processing an Incoming Request, Section 5.2.1 in Page 26-27, this is inherent since there must be at least a subsystem to handle the Incoming Request) at each of a first component and a second component, each of said handlers supporting a corresponding one of said plurality of types and versions of requests;

sending a request (sending requests, 2nd line of Section 4.1.1 in Page 15) from the first component (a command generator, 1st line of Section 4.1.1 in Page 15), to the second component (Command Responder, 1st line of Section 4.1.2 in Page 15), said sent request having a particular identifier (e.g., messageProcessingModel, first bullet of Page 27) that is one of the plurality of type-version identifiers indicating a particular type and version of said sent request; (0 for SNMPv1, or 1 or SNMPv2c of Section 5.2.1 in Page 27) and

causing said second component to,

extract said particular identifier (inherent from Section 5.2.1 in Page 26-27) of said sent request,

use said particular identifier (e.g., messageProcessingModel, first bullet of Page 27) to determine whether one of said handlers installed at said second component properly supports said particular type and version of said sent request , and

if a proper one of said installed handlers supports said particular type and version of said sent request, using said proper handler to process said sent request (Command Responder of 4.1.2, including all it's subsections).

As to **claim 2**, RFC 2576 discloses a method as recited in claim 1, further comprising:

providing said first component with a first data structure (a local database, 3rd line of 4.1.1 in page 15) for indicating whether or not said first component may send requests of said particular type and version to said second component; and

before said step of sending a request, accessing said first data structure to determine whether requests of said particular type and version may be sent to said second component (select the appropriate message version, 3rd line of 4.1.1 in page 15).

As to **claim 5**, RFC 2576 discloses a method as recited in claim 1, wherein said sent request carries said particular identifier in a header field, the method further comprising:

extending said particular identifier by defining a sub-type-version identifier (sub-identifiers, item 5 of Page 9) in said header field, said particular identifier having a value that indicates the presence of said sub-type-version identifier in said header field (item 5 of Page 9).

As to **claim 6**, RFC 2576 discloses a method as recited in claim 5, the method further comprising:

extending said sub-type-version identifier by defining a sub-sub-type-version identifier in said header field, said sub-type-version identifier having a value that indicates the presence of said sub-sub-type-version identifier in said header field (Section 2.1.1 of Page 9, with version identifier being for the version of SNMP, sub-type being for OBJECTS or Variables, and sub-sub-type be for attributes of OBJECTS or Variables).

As to **claim 7**, RFC 2576 discloses a method as recited in claim 1, wherein said step of using said particular identifier to determine whether one of said handlers installed at said second component properly supports said particular type and version of said sent request further comprises:

using said particular identifier to index a second data structure having a plurality of pointers (multi-lingual implementation, Lines 3-5 Section 4 of Page 14), each said pointer being associated with one of said plurality of type-version identifiers and pointing to a corresponding one of said handlers installed at said second component (Command Responder in Section 4.1.2, Page 15 and multi-lingual implementation, Lines 3-5 Section 4 of Page 14).

As to **claim 8**, RFC 2576 discloses a method as recited in claim 7, further comprising:

installing an unsupported-type-version at said second component, said unsupported-type-version handler supporting requests carrying any one of a plurality of

Art Unit: 2609

unsupported ones of said type-version identifiers wherein said sent request carries one of said unsupported type-version identifiers handler (status-error, particularly authorizationError or noSuchName, Section 4.3, Page 24); and

invoking said unsupported-type-version handler in response to said received unsupported type-version identifier without indexing said second data structure (Section 4.3, Page 24, where status-error messages is used to handle requests that are not unsupported);

whereby said second data structure need not store pointers for each of said unsupported type-version identifiers (Section 4.3, Page 24, where since status-error messages is used to handle requests that are not unsupported, there is no need to use different pointers for each of said unsupported type-version identifiers).

For **Claim 9**, RFC 2576 discloses a versioning infrastructure for minimizing compatibility issues among interacting components of different dialect versions, comprising:

a plurality of components between which requests may be exchanged , each request being of a type and version from among a plurality of types and versions and having a header carrying a type-version identifier indicating a corresponding type and version of said request (multi-lingual implementation, Section 4.1, Page 15); and

each said component including,

an input port (command responder, Section 4.1.2, Page 15) for receiving one of said requests,

at least one handler (multi-lingual implementation, Section 4.1, Page 15)

Art Unit: 2609

supporting requests of a corresponding one of the plurality of types and versions,

a pointer array (lingual modules for multi-lingual implementation, Section 4.1, Page 15) having a plurality of elements each being a pointer to a corresponding one of said handlers, and

switching logic operable to extract said type-version identifier carried by said received request (logic presented in the first line of Section 4.1.2, Page 15),

use said extracted type-version identifier to index said pointer array to determine a selected one of said handlers, and invoke said selected handler (e.g., messageProcessingModel and securityModel first 2 bullets, Page 27).

As to **Claim 10**, RFC 2576 discloses a versioning infrastructure as recited in claim 9, wherein each said component further includes installation logic operable to install said handlers (multi-lingual implementation, Section 4.1, Page 14) at said component.

As to **Claim 12**, RFC 2576 discloses a versioning infrastructure as recited in claim 9, wherein each said component further includes incompatibility reporting logic operable to report receipt of a request that is not supported by any one of said installed handlers (status-error, particularly authorizationError or noSuchName, Section 4.3, Page 24).

As to **Claim 17**, they are rejected for the sam

As to **Claim 17** RFC 2576 discloses a versioning infrastructure as recited in claim 9, wherein said header of each said request carries a sub-type-version identifier (sub-identifiers, item 5 of Page 9) for extending said type-version identifier, said type-version identifier having a value that indicates the presence of said sub-type-version identifier (item 5 of Page 9).

As to **Claim 18**, RFC 2576 discloses a versioning infrastructure as recited in claim 17, wherein said header of each said request carries a sub-sub-type-version identifier (sub-identifiers, item 5 of Page 9) for extending said sub-type-version identifier, said sub-type-version identifier having a value that indicates the presence of said sub-sub-type-version identifier (Section 2.1.1 of Page 9, with version identifier being for the version of SNMP, sub-type being for OBJECTS or Variables, and sub-sub-type be for attributes of OBJECTS or Variables).

As to **Claim 19**, RFC 2576 discloses a versioning infrastructure as recited in claim 9, wherein one of said handlers is an unsupported-type-version handler, said unsupported-type-version handler supporting requests carrying any one of a plurality of unsupported ones of said type-version identifiers (status-error, particularly authorizationError or noSuchName, Section 4.3, Page 24); and

said switching logic includes memory saving logic that invokes said unsupported-type-version handler in response to one of said unsupported-type-version identifiers carried by said received request without indexing said pointer array whereby said pointer array need not store pointers for each of said unsupported type-version identifiers (inherent from Section 4.3, Page 24, where status-error messages is used to handle requests that are not unsupported).

For **Claim 20**, it is a means for version of claim 9 therefore is rejected for the same reasons as explained in claim 9 above.

For **Claim 21**, it is the corresponding computer readable medium claim of method claim 1, therefore are rejected for the same reasons as shown above.

Art Unit: 2609

For **Claim 22**, they are corresponding protocol claims of claims 1, 9, 20, or 21, therefore are rejected for the same reasons as shown above.

For **Claim 23**, they are corresponding interface claims of claims 1, 9, 20, or 21, therefore are rejected for the same reasons as shown above.

For **Claim 24**, it is corresponding network claim of method claim 1, therefore, is rejected for the same reasons as explained in claim 1 above.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

6. **Claims 3-4, 11, 13, 14-16** are rejected under 35 U.S.C. 103(a) as being unpatentable over RFC 2576.

As to **Claim 3**, RFC 2576 discloses a method of claim 2;

RFC 2576 is **silent on** explicitly one identifier for indicating a stop sending control request.

Art Unit: 2609

However, RFC 2576 teaches status-error type-version identifier, particularly authorizationError or noSuchName (Section 4.3, Page 24) that the second component can use as a stop sending request to inform the first component that the type-version is not supported.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to let said second component to use status-error type message to function as a stop sending control request to said first component that it should stop sending requests of said particular type and version to said second component for the benefit of avoiding unnecessary traffic.

As to **Claim 4**, RFC 2576 discloses a method of claim 3, further comprising:
receiving said stop sending control request at said first component (Section 4.3, Page 24, Error-status message is received at said first component); and
updating said first data structure to indicate that said first component may not send requests of said particular type and version to said second component

(for the same reasons as explained in claim 3 above).

As to **Claim 11**, RFC 2576 discloses a versioning infrastructure as recited in claim 9; but is **silent on** explicitly disclosing wherein each said component further includes a flag array having at least one flag element corresponding to an intended receiver component and to a particular one of the plurality of types and versions, each said flag element indicating whether or not said component may send requests of said corresponding particular type and version to said corresponding intended receiver component;

however, RFC 2576 teaches multi-lingual implementation (Lines 3-5 of Section 4 of Page 14) where each lingual supports a specific type and version; this is equivalent to a flag of array which contains information of which components are supported and which are not and flag array are commonly used in the art for this purpose.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to use a flag array to indicate which components are supported and which are not for the benefit of providing clear picture of system configuration.

As to **Claim 13**, RFC 2576 discloses a versioning infrastructure as recited in claim 9; but is **silent on** explicitly disclosing wherein said requests include at least one control request for managing versioning in accordance with the infrastructure, each said control request carrying a corresponding control type-version identifier specifying a type of control request, said control requests including a stop sending control request to be sent by a notifying one of the components to a receiving one of the components to indicate that said receiving component should stop sending requests of a particular one of the

Art Unit: 2609

plurality of types and versions to said notifying component.

However, RFC 2576 teaches status-error type-version identifier, particularly authorizationError or noSuchName (Section 4.3, Page 24) that the second component can use as a stop sending request to inform the first component that the type-version is not supported.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to let said second component to use status-error type message to function as a stop sending control request to said first component that it should stop sending requests of said particular type and version to said second component for the benefit of avoiding unnecessary traffic.

As to **Claim 14**, RFC 2576 discloses a versioning infrastructure as recited in claim 13 but is **silent on** explicitly disclosing wherein said control requests further include a start sending control request to be sent by a notifying one of the components to a receiving one of the components to indicate that said receiving component may start sending requests of a particular one of the plurality of types and versions to said notifying component.

However, RFC 2576 teaches status-error type-version identifier, particularly authorizationError or noSuchName (Section 4.3, Page 24) that the second component can use as a stop sending request to inform the first component that the type-version is not supported.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to let said second component to use status-error type message to

Art Unit: 2609

function as a stop sending control request to said first component that it should stop sending requests of said particular type and version to said second component for the benefit of avoiding unnecessary traffic.

As to **Claim 15**, RFC 2576 discloses a versioning infrastructure as recited in claim 13; but is **silent on** explicitly disclosing wherein said control requests further include a test connection request to be sent by a notifying one of the components to send a test connection response to said notifying component so as to probe an underlying connection between said notifying component and said receiving component.

However, RFC 2576 teaches getRequest message (Section 4.1.2.3 of Page 17) that can be used to test underlying connection (besides being used to collecting information) between said notifying component and said receiving component (this is commonly used in the art; official notes).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention getRequest message to test underlying connection for the benefit of collecting network operation status.

As to **Claim 16**, RFC 2576 discloses a versioning infrastructure as recited in claim 13; but is **silent on** explicitly disclosing wherein said control requests further include a message reporting request to be sent by a notifying one of the components that has received an unrecognized request to a receiving one of the components to indicate that said receiving component may send a message reporting response that has a message describing said unrecognized request to said notifying component.

Art Unit: 2609

However, RFC 2576 teaches status-error type-version identifier, particularly authorizationError or noSuchName (Section 4.3, Page 24) that the second component can use as a stop sending request to inform the first component that the type-version is not supported.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to let said second component to use status-error type message to to notify sender that one of components in the message is not supported for the benefit of efficient communication between the sender and receiver.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jianye Wu whose telephone number is (571)270-1665. The examiner can normally be reached on Monday to Friday, 8am to 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eliseo Ramos-Feliciano can be reached on (571)272-7925. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2609

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

 5/14/07
ELISEO RAMOS-FELICIANO
SUPERVISORY PATENT EXAMINER